整数線形計画問題の解非存在性判定を利用した通信プロトコルの自動検証について(計算アルゴリズムと計算量の基礎理論)

AUTHOR(S):

東野, 輝夫; 谷口, 健一

CITATION:

東野, 輝夫 ...[et al]. 整数線形計画問題の解非存在性判定を利用した通信プロトコルの自動検証について(計算アルゴリズムと計算量の基礎理論). 数理解析研究所講究録 1989, 695: 243-252

ISSUE DATE:

1989-06

URL:

http://hdl.handle.net/2433/101384

数理解析研究所講究録
695巻 1989年 243-252

整数線形計画問題の解非存在性判定を利用した通信プロトコルの自動検証について

大阪大学基礎工学部情報工学科　東野輝夫（ Teruo Higashino ）
大阪大学基礎工学部情報工学科　谷口健一（ Ken'ichi Taniguchi ）

1．まえがき

近年のデータ通信の発展に伴い，通信の手順を規定する通信プロトコル（以下単にプロトコルと呼ぶ）の厳密かつ形式的な記述やその正しさの検証は，重要な課題になっている．プロトコルにおける検証の問題は大きく分けると，安全性の検証と生存性の検証に分類される．前者は，いわゆる部分正当性の証明に相当し，与えられた性質が不変式として成り立つかどうかの証明に帰着される．また，後者は，全正当性の証明に相当し，いつかは与えられた性質が成り立つ（あるいは，与えられた性質が何回も成り立つ）ことの証明に帰着される．従来，プロトコルの形式的な記述法としては，(1) 有限オートマトンやペトリネットモデルを用いる方法や，(2) 高級言語を用いる方法等が提案されている[1]．前者は，到達可能性等生存性の検証を機械的に行うためのモデルとしては適当であるが，パラメータ値が取り扱えないため，安全性の検証を行うためのモデルとして不適当である．又，後者は，安全性や生存性の検証を行うためのモデルとしては適当であるが，それらの検証を機械的に行うための有効な方法があまり提案されていない．本論文では，安全性と生存性の検証を機械的に行う上で適当と思われるモデルを一つ定め，そのモデル上で２つの性質を機械的に証明するための一つの方法を提案する[2,3]．

2．プロトコルマシンのモデル

本論文では，プロトコルマシンを「有限制御部」及び整数値を保持する有限個の「レジスタ」を持つオートマトンとしてモデル化する．状態数は有限とし，各状態には適当な整数値を割り当てる．また，「相手局からのフレームの受信」や「タイマーからの割り込み」，「相手局へのデータの送信」，あるいは，「相手局へのデータの再送」などを各々，このオートマトンの「入力記号」に対応付ける．入力記号はその動作の種類により，(1) 相手局へのメッセージの送信動作に相当する入力記号の有限集合（送信動作集合）I_s，(2) 相手局からのメッセージの受信動作に相当する入力記号の有限集合（受信動作集合）I_r，(3) それ以外の動作に相当する入力記号の有限集合（単独動作集合）I_o，に分ける．受信動作集合 I_r に属する入力記号のみ「パラメータ」を持ってもよい．パラメータは１個の整数とし，通信相手局から受信するメッセージに対応付ける．また，送信動作集合 I_s に属する入力記号のみ出力を定義する．出力も整数値とし，通信相手局へ送信するメッセージに対応付ける．

プロトコルマシンは，現在の状態（以下，状態は有限制御部の状態を表す），レジス

タ値及び入力（入力記号とパラメータ値の対）の組に対して，次の状態，次のレジスタ値，出力が定義されるいわゆるMealy 型のオートマトンとしてモデル化する．

モデル化したプロトコルマシンの仕様は，図１，図２（例１，例２参照）のようなグラフの形で指定する．

図１において，グラフの頂点はオートマトンの状態に対応し，有向辺によって各入力に対する次の状態，次のレジスタ値，出力を指定する．

グラフの各辺ｓ→ｓ’には，ｓ→ｓ’がI_s，I_r，I_oの何れに属するかに応じて各々４字組のラベル

① $<\alpha, c(R_n), \delta(R_n), \rho(R_n)>$

② $<\alpha(p), c(R_n, p), \delta(R_n, p), *>$

③ $<\alpha, c(R_n), \delta(R_n), *>$

が付加されている．ここで，R_n はプロトコルマシンのレジスタ値を表す変数の組を表し，ｐは入力記号αのパラメータ値を表す変数である．$c(R_n)$，$c(R_n, p)$の値域はブール値とし，$\delta(R_n)$，$\delta(R_n, p)$ の値域は整数のn 字組とする．又，$\rho(R_n)$ の値域は整数である．以下，変数の線形結合をＰ項，線形不等式の論理結合をＰ文と呼ぶ．本論文では，安全性や生存性を機械的に検証するため，$c(R_n)$や$c(R_n, p)$はR_n（またはR_n, p）を変数とするＰ文で記述されるとし，$\delta(R_n)$ や$\delta(R_n, p)$ は，R_n（またはR_n, p）を変数とするＰ項のn 字組で記述されるとする．また$\rho(R_n)$ はR_n を変数とするＰ項で記述されるとする．

いま，現在のレジスタ値の組がN_n で入力がαの時，もし，$c(N_n)$の値が真ならば，次の状態はｓ’であり，次のレジスタ値は$\delta(N_n)$である．また，その遷移に対する出力として$\rho(N_n)$が出力される．なお，ｓから４字組のラベルの第１成分がαであるような有向辺が必ず存在する必要はないが（次の状態，次のレジスタ値，出力が未定義でもよいが），そのような有向辺が１本でも存在する場合，任意の入力α及びレジスタ値N_n に対して，$c(N_n)$の値を真にするような有向辺が唯一つ存在すると仮定する．尚，②，③では出力が空であるので，ラベルの第４成分を＊と記述している．

［例１　プロトコルマシンM1］

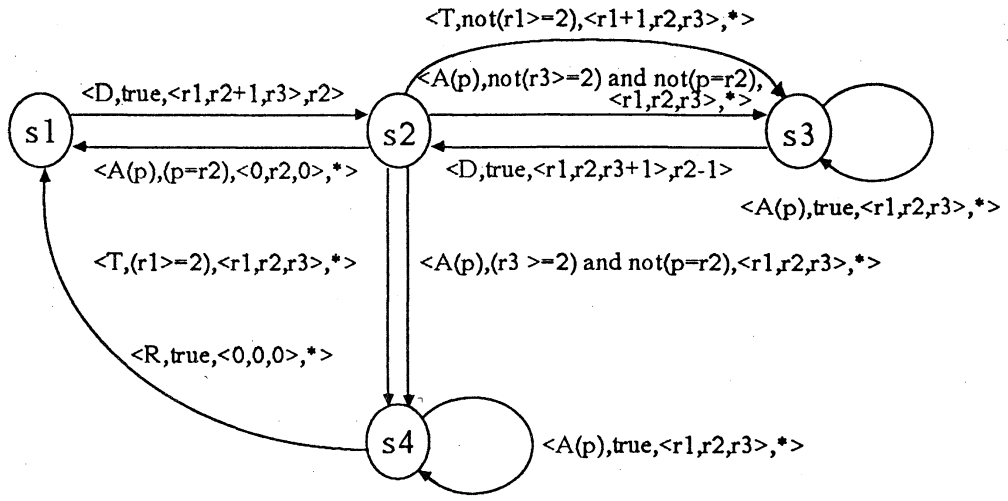
図１のプロトコルマシンM1はデータリンクレベルにおけるHDLC手順の送信局を簡単にしたものである．例えば，現在の状態がs_2で，レジスタR_1, R_2, R_3 の値が各々r_1, r_2, r_3で，入力がＴの場合，r_1が２以上ならばs_4 に遷移し，レジスタR_1，R_2，R_3の値は各々r_1, r_2, r_3 のままであり，r_1 が２未満ならばs_3に遷移し，R_1, R_2, R_3 の値は各々r_1+1, r_2, r_3 となる．■

［例２　プロトコルマシンM2］

図２のプロトコルマシンM2はデータリンクレベルにおけるHDLC手順の受信局を簡単にしたものである．■

３．プロトコルマシン対の動作

本論文では，２つのプロトコルマシン間の通信回線にはバッファがなく，伝送遅延が



状態　ｓ１：初期状態（データ未送信状態）
ｓ２：データ送信済状態
ｓ３：データ未再送状態
ｓ４：オペレータ通告状態

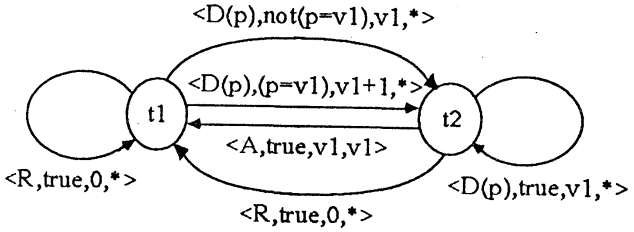
入力記号　Ｄ：相手局へのデータ送信
Ｒ：オペレータによる相手局への
回復終了の通知命令
Ａ：相手局からのデータ受信
Ｔ：タイマからのタイムアウト通知受信

レジスタ　Ｒ１：タイムアウト通知の連続受信回数を保持
（初期値　０）
Ｒ２：異なるデータの送信された数を保持
（初期値　０）
Ｒ３：同一データを再送した回数を保持
（初期値　０）

Ｉｓ＝｛Ｄ,Ｒ｝,Ｉｒ＝｛Ａ｝,Ｉｏ＝｛Ｔ｝

ｒ１：現在のレジスタｒ１の値を表す変数
ｒ２：現在のレジスタｒ２の値を表す変数
ｒ３：現在のレジスタｒ３の値を表す変数
ｐ　：Ａの持つパラメータの値を表す変数

図１　　送信局の仕様



状態　ｔ１：初期状態（データ受信待状態）
ｔ２：応答未送信状態

入力記号　Ａ：相手局への応答送信
Ｄ：相手局からのデータ受信
Ｒ：相手局からの回復通知受信

Ｉｓ＝｛Ａ｝,Ｉｒ＝｛Ｄ，Ｒ｝,Ｉｏ＝ϕ

レジスタ　Ｖ１：異なるデータの受信した数を保持
（初期値　０）
ｖ１：現在のレジスタＶ１の値を表す変数
ｐ　：Ｄの入力パラメータの値を表す変数

図２　　受信局の仕様

ないと仮定する．また，通信回線上での伝送誤りは，そのパケットが通信回線上で消失したものとみなす．このため，互いに通信を行う二つのプロトコルマシンM1，M2の動作を次のように定める．但し，以下ではI_{s1}，I_{s2}はM1，M2の送信動作集合を，I_{r1}，I_{r2}は受信動作集合を，I_{o1}，I_{o2}は単独動作集合を各々表すとする．また，M1(M2)の送信動作αに対応するM2(M1)の受信動作もαで表す．すなわち，$I_{s1}=I_{r2}$，$I_{r1}=I_{s2}$が成り立つと仮定する．

［プロトコルマシン対の動作］

(1) M1の送信動作$\alpha\in I_{s1}$と，αに対応するM2の受信動作$\alpha\in I_{r2}$が同時に行われる（正常受信に相当）．

(2) M1の送信動作$\alpha\in I_{s1}$のみが単独で行われる（通信回線上でのパケットの消失に相当）．

(3) M2の送信動作$\beta\in I_{s2}$と，βに対応するM1の受信動作$\beta\in I_{r1}$が同時に行われる（正常受信に相当）．

(4) M2の送信動作$\beta\in I_{s2}$のみが単独で行われる（通信回線上でのパケットの消失に相当）．

(5) I_{o1}に属するM1の動作γが単独で行われる．

(6) I_{o2}に属するM2の動作γが単独で行われる．

本論文では，(1)～(6)の動作を各々[α, α], [α, *], [β, β], [*, β], [γ, *], [*, γ]で表す．また，送受信動作[α, α]については，M1から送信動作αによって出力される整数値がM2の受信動作αの入力パラメータ値として割り当てられると考える（[β, β]についても同様）． ■

次に(1)～(6)の動作によってM1，M2の状態やレジスタ値がどのように変化するかについて述べる．

［各プロトコルマシンの状態やレジスタ値の変化］

(A) M1, M2が到達した状態の２字組$\langle s, t\rangle$において，M1で$<\alpha, c(\mathbb{R}_n), \delta(\mathbb{R}_n), \rho(\mathbb{R}_n)>$なるラベルを持つ有向辺s→s'が存在し（但し$\alpha$は送信動作を表す入力記号），M1のレジスタ値の組$\mathbb{N}_n$に対して，$c(\mathbb{N}_n)$が真であれば，そのとき，動作[$\alpha$, *]は実行可能であり，M1の状態がs'に変わり，レジスタ値が$\delta(\mathbb{N}_n)$に変化する．M2の状態やレジスタ値は変化しない（動作[*, β]についても同様，M2の状態とレジスタ値のみが変化する）．

(B) M1，M2が到達した状態の２字組$\langle s, t\rangle$において，M1で$<\alpha, c_1(\mathbb{R}_n), \delta_1(\mathbb{R}_n), \rho_1(\mathbb{R}_n)>$なるラベルを持つ有向辺s→s'が（但し$\alpha$は送信動作を表す），M2で$<\alpha(p), c_2(\nabla_m, p), \delta_2(\nabla_m, p), *>$なるラベルを持つ有向辺t→t'が各々存在し（但し$\alpha(p)$は，M1の送信動作$\alpha$に対応するM2の受信動作），M1のレジスタ値の組$\mathbb{N}_n$に対して，$c_1(\mathbb{N}_n)$が真でM2のレジスタ値の組$\mathbb{C}_m$とM1の出力$\rho_1(\mathbb{N}_n)$をc2の変数$\nabla_m$, pに代入した式$c_2(\mathbb{C}_m, \rho_1(\mathbb{N}_n))$の値が真であれば，その時，動作[$\alpha$, α]は実行可能であり，M1, M2の状態は各々s', t'に変化し，レジスタ値が各々$\delta_1(\mathbb{N}_n)$, $\delta_2(\mathbb{C}_m, \rho_1(\mathbb{N}_n))$に変化する（[$\beta$, β]についても同様）

(C) [γ,*],[*, γ] を実行したときの状態やレジスタ値の変化は[α,*],[*, β] を実行したときの状態やレジスタ値の変化と同じ. ■

例えば，図1のプロトコルマシンM1の状態がs_1で，図2のプロトコルマシンM2の状態がt_2であるとする．M1では$s_1 \rightarrow s_2$なる有向辺が存在し，その有向辺のラベルの第1成分はDで，第2成分がtrueである．このため，$\langle s_1, t_1 \rangle$ において，動作[D,*]（M1がデータの送信を行ったがそのデータが回線上で消失したことを表す）が実行可能であり，実行後は，M1の状態のみがs_2に変化し，M1のレジスタR_2 の値が1増加する．

また，図2のプロトコルマシンM2では，ラベルの第1成分がDであるような有向辺$t_1 \rightarrow t_2$が存在するので動作[D, D] が実行可能である．動作[D, D] の実行時にM1のレジスタR_2 の値がM2の受信動作Dの入力パラメータ値として割り当てられる．もし，この値がM2のレジスタR_4 の値と等しければ，動作[D, D] の実行後，M2は状態t_2に遷移し，レジスタR_4 の値は1増加する．R_2 の値とR_4 の値が等しくなければR_4 の値は変化しない．尚，動作[D, D] の実行後のM1の状態やレジスタ値は，[D,*]実行後の各値と同じである．

M1，M2は，各々の初期状態の2字組から遷移を開始し，(1)～(6)の動作によって，(A)～(C) のような状態（レジスタ値）変化を行いながら遷移を繰り返す．

以下では，(1)～(6)のような動作の系列をM1，M2に対する入力系列と呼ぶ．M1，M2に対する入力系列ωに対してσ(M1, ω)，σ(M2, ω) はωを実行後に到達するM1，M2の状態を表し，δ(M1, ω)，δ(M2, ω) はωを実行後に定まるM1，M2のレジスタ値の組を表すとする．

尚，以下では(A)や(C) のような動作に対して，

$$\langle s, t, \mathbb{R}_n, \nabla_m \rangle \Rightarrow \langle s', t, \delta_1(\mathbb{R}_n), \nabla_m \rangle$$

をM1，M2の遷移対と呼び，$\langle [\alpha, *], c_1(\mathbb{R}_n) \rangle$をそのラベルとする．同様に，(B) のような動作に対しても，

$$\langle s, t, \mathbb{R}_n, \nabla_m \rangle \Rightarrow \langle s', t', \delta_1(\mathbb{R}_n), \delta_2(\nabla_m) \rangle$$

をM1，M2の遷移対と呼び，$\langle [\alpha, \alpha], \{c_1(\mathbb{R}_n)$ and $c_2(\nabla_m, \rho_1(\mathbb{R}_n))\} \rangle$をそのラベルとする．M1，M2の遷移対は，現在の状態・レジスタ値とラベルの第1成分の動作実行後の状態・レジスタ値の間の関係を表している．また，ラベルの第2成分はその動作が実行可能であるための条件を表している．

4．安全性の検証法

以下$\Phi(u_1, u_2, \mathbb{R}_n, \nabla_m)$ はM1とM2の状態とレジスタ値を表すP文とする（但し，u_1, u_2はM1, M2の状態を表す変数とし，$\mathbb{R}_n$はM1の∇_mはM2のレジスタ値を表す変数の組とする）．任意の入力系列ωに対して，

$$\Phi(\sigma(M1, \omega), \sigma(M2, \omega), \delta(M1, \omega), \delta(M2, \omega))$$

が真である時，ΦはM1，M2に対する不変式であると呼ぶ．本論文では，上述の不変式をプロトコルの安全性と考える．

例えば，図1，図2のプロトコルマシンに対して，

$\Phi'(u_1, u_2, r1, r2, r3, v1) = (0 \leqq r1 \text{ and } r1 \leqq 2)$

とすれば，Φ' は不変式となる．

一般にΦ' のような不変式の証明は決定不能であるので，本論文では，Φ' のような不変式を機械的に証明するための一つの方法（十分条件）について述べる．

本論文のモデルでは，

$$\langle s, t, \mathbb{R}_n, \nabla_m \rangle \Rightarrow \langle s', t', \delta_1(\mathbb{R}_n), \delta_2(\nabla_m) \rangle$$

なる遷移対（そのラベルを$\langle [\alpha, \alpha], \{c_1(\mathbb{R}_n) \text{ and } c_2(\nabla_m, \rho_1(\mathbb{R}_n))\} \rangle$とする）に対して，$\delta_1(\mathbb{R}_n), \delta_2(\nabla_m)$はそれぞれ変数$\mathbb{R}_n, \nabla_m$ の線形結合で表される．このため，P文Φに対して，

$$[\Phi(s, t, \mathbb{R}_n, \nabla_m) \supset \text{not}\{c_1(\mathbb{R}_n) \text{ and } c_2(\nabla_m, \rho_1(\mathbb{R}_n))\}]$$

及び

$$[\Phi(s, t, \mathbb{R}_n, \nabla_m) \text{ and } \{c_1(\mathbb{R}_n) \text{ and } c_2(\nabla_m, \rho_1(\mathbb{R}_n))\} \supset \Phi(s', t', \delta_1(\mathbb{R}_n), \delta_2(\nabla_m))]$$

は，$\mathbb{R}_n, \nabla_m$ を変数とするP文である．よって

$$\forall \mathbb{R}_n, \nabla_m [\Phi(s, t, \mathbb{R}_n, \nabla_m) \supset \text{not}\{c_1(\mathbb{R}_n) \text{ and } c_2(\nabla_m, \rho_1(\mathbb{R}_n))\}] \quad \text{--(*1)}$$

及び

$$\forall \mathbb{R}_n, \nabla_m [\Phi(s, t, \mathbb{R}_n, \nabla_m) \text{ and } \{c_1(\mathbb{R}_n) \text{ and } c_2(\nabla_m, \rho_1(\mathbb{R}_n))\} \supset \Phi(s', t', \delta_1(\mathbb{R}_n), \delta_2(\nabla_m))] \quad \text{--(*2)}$$

は変数$\mathbb{R}_n, \nabla_m$ が全称作用素$\forall$で束縛されたプレスブルガー文となり，(*2),(*3) の真偽は何れも決定可能である[4,5]．(*1)が成り立つ時，その遷移対はタイプⅠであるといい，タイプⅠでなく(*2)が成り立つ時，その遷移対はタイプⅡであるという．遷移対がタイプⅠでもタイプⅡでもない場合，その遷移対はタイプⅢであるという．遷移対がタイプⅠならば，状態s, t 及びΦを満たすレジスタ値の組に対して，その遷移対で書かれた遷移は実行されない．遷移対がタイプⅡならば，状態s, t 及びΦを満たすレジスタ値の組に対して，その遷移が実行された後の状態・レジスタ値の組もΦを満足する．以下，$\langle s, t, \mathbb{R}_n, \nabla_m \rangle \Rightarrow \langle s', t', \delta_1(\mathbb{R}_n), \delta_2(\nabla_m) \rangle$に対して，$\langle s, t \rangle, \langle s', t' \rangle$ を各々この遷移対の親，子とよぶ．

集合$\{\langle \sigma(M1, \omega), \sigma(M2, \omega) \rangle\}$ に属する状態対を親とする遷移対すべてがタイプⅠまたはタイプⅡならば，Φが不変式として成り立つ．よって，次のような手続きSAFENESSを実行し，「成功」が出力されれば，Φが不変式として成り立つ．

［安全性の検証アルゴリズム］

```
procedure SAFENESS(M1, M2, Φ) ;
  S←⟨s_I, t_I⟩ ; /* s_I, t_I はM1, M2の初期状態，⟨s_I, t_I⟩はﾏｰｸされていない */
  while S中でﾏｰｸされていない状態対が存在 do
   begin
     ⟨s, t⟩をS中でﾏｰｸされていない状態対とする
     if ⟨s, t⟩が親であるような遷移対の中でﾀｲﾌﾟⅢの遷移対が存在
        then print(失敗) ;
```

〈s,t〉をﾏｰｸする
S←S∪{〈s',t'〉|〈s',t'〉がS中でﾏｰｸされておらず，且つ〈s,t〉が親で
〈s',t'〉がその子であるようなﾀｲﾌﾟⅡの遷移対が存在する}
end ;
Ω←S ;
print(成功) ;
end.

(証明) 略 ■

上述のΦ'に対して，手続きSAFENESSを用いてΦ'が不変式であることを証明できる．尚，手続きが「成功」を出力した場合，

$$\{\langle \sigma(M1,\omega), \sigma(M2,\omega)\rangle\}^{\neg} \subset \Omega$$

が成り立つ．

5．生存性の検証法

本論文では，与えられた性質Φが何回も成り立つことをプロトコルの生存性と考える．すなわち，生存性を次のように定義する．

[生存性]

M1，M2の初期状態と初期レジスタ値の組から遷移を行うことによって到達可能な任意の状態とレジスタ値の組がP文Φを満足すれば，それ以降どのような遷移を行っても，いつかはΦを満足する状態とレジスタ値の組に遷移できること． ■

一般に異なるΦを指定することによって，異なる生存性を議論できる．例えば，図1，2のM1，M2に対して，Φとして

$$\Phi''(u_1,u_2,r_1,r_2,r_3,r_4)=(u_1=s_1)\ \text{and}\ (u_2=t_1)$$

を指定すれば，「初期状態の2字組へ，どの状態の2字組からも復帰できる」かどうかが議論できる．一般に，M1，M2及びP文Φが与えられて，M1，M2がΦに対する生存性を持つかどうかは決定不能である．本論文では，Φ''のような生存性を機械的に証明するための一つの方法（十分条件）について述べる．

[P文Φ及び頂点$\langle s,t,\mathbb{R}_n,\nabla_m\rangle$に対するk-到達可能木$T_k(\langle s,t,\mathbb{R}_n,\nabla_m\rangle,\Phi)$]

k-到達可能木$T_k(\langle s,t,\mathbb{R}_n,\nabla_m\rangle,\Phi)$では，頂点$\langle s,t,\mathbb{R}_n,\nabla_m\rangle$を根とする．もし，$\langle s,t,\mathbb{R}_n,\nabla_m\rangle \Rightarrow \langle s',t',\delta_1(\mathbb{R}_n),\delta_2(\nabla_m)\rangle$なる遷移対が存在すれば，$\langle s',t',\delta_1(\mathbb{R}_n),\delta_2(\nabla_m)\rangle$を$\langle s,t,\mathbb{R}_n,\nabla_m\rangle$の子供とする．次に，子供の頂点に対して$\langle s',t',\mathbb{R}_n,\nabla_m\rangle \Rightarrow \langle s'',t'',\eta_1(\mathbb{R}_n),\eta_2(\nabla_m)\rangle$なる遷移対が存在すれば，頂点$\langle s'',t'',\eta_1(\delta_1(\mathbb{R}_n)),\eta_2(\delta_1(\nabla_m))\rangle$をその子供とする．この操作を根からの距離がkになるまで繰り返す．また，各枝にはラベルを付加し，そのラベルはその枝を生成する遷移対のラベル$\langle [a,b],c(\mathbb{R}_n,\nabla_m)\rangle$と同じとする．この時，根から到達可能な頂点$\langle s_i,t_i,\mathbb{R}'_n,\nabla'_m\rangle$に対して$\langle [a,b],c(\mathbb{R}_n,\nabla_m)\rangle$なるラベルを持つ枝が存在し，且つ指定されたΦに対して，

$$\forall \mathbb{R}_n,\nabla_m[\Phi(s,t,\mathbb{R}_n,\nabla_m)\supset \text{not}(c(\mathbb{R}'_n,\nabla'_m))]$$

が真（s,tでΦを満足すれば，s_i,t_iに遷移した時，この遷移が実行されない）ならば，

その枝及びその枝以下の部分木を削除する．このようにして生成された木を $T_k(\langle s, t, \mathbb{R}_n, \nabla_m\rangle, \Phi)$とする．すなわち，$T_k(\langle s, t, \mathbb{R}_n, \nabla_m\rangle, \Phi)$では根$\langle s, t, \mathbb{R}_n, \nabla_m\rangle$の時点でのレジスタ値を表す変数の組$\mathbb{R}_n, \nabla_m$に対して，各頂点のレジスタ値が$\mathbb{R}_n, \nabla_m$を用いてどのような一次結合で表されるかを示している． ■

［例３　2-到達可能木］

図３は，上述のＰ文Φ”及び図１，図２で定義したマシンM1，M2における頂点$\langle s_1, t_1, \mathbb{R}_n, \nabla_m\rangle$に対する2-到達可能木である． ■

［補題１］

k-到達可能木$T_k(\langle s_0, t_0, \mathbb{R}_n, \nabla_m\rangle, \Phi)$の根から葉へ至るすべての有向道に対して，その有向道上のいずれかの頂点$\langle s_h, t_h, \mathbb{R}'_n, \nabla'_m\rangle$（$\mathbb{R}'_n, \nabla'_m$は変数$\mathbb{R}_n, \nabla_m$の一次結合）が

$$\forall \mathbb{R}_n, \nabla_m\ [\Phi(s_0, t_0, \mathbb{R}_n, \nabla_m) \supset \Phi(s_h, t_h, \mathbb{R}'_n, \nabla'_m)] \qquad \cdots(a)$$

を満足するとする．この時，状態の２字組$\langle s_0, t_0\rangle$においてレジスタ値の組がΦを満足すれば，そこから高々ｋ回の動作によって到達する状態の２字組と，そこまでで変化したレジスタ値の組がΦを満足する． ■

以下では，k-到達可能木$T_k(\langle s_0, t_0, \mathbb{R}_n, \nabla_m\rangle, \Phi)$が補題１を満足するとき，$\langle s_0, t_0\rangle$がΦに対するk-生存性を持つという．また，(a)を満足するような状態の２字組$\langle s_h, t_h\rangle$の集合を$F_k(\langle s_0, t_0\rangle, \Phi)$で表す．

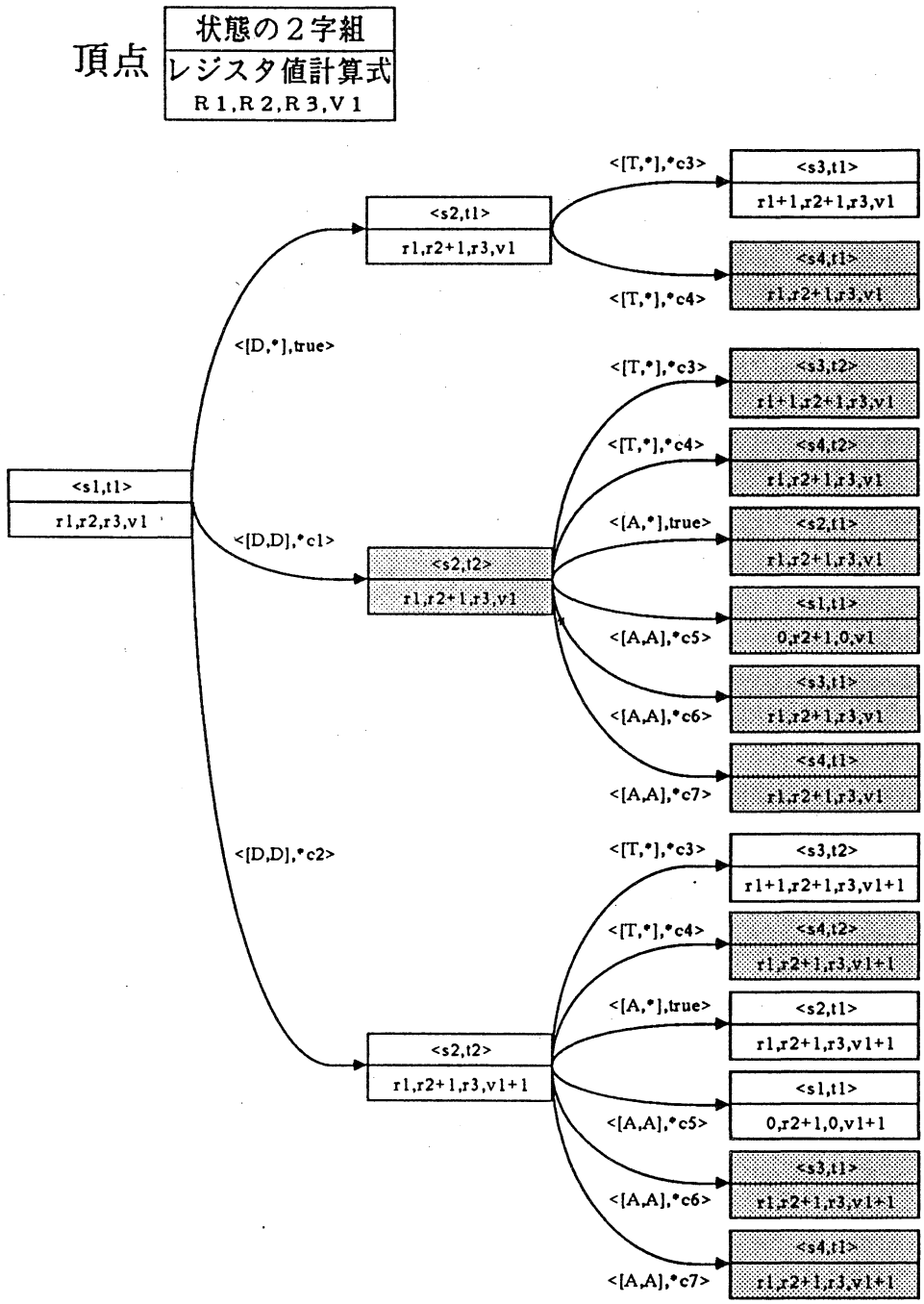
以上より，M1,M2，Ｐ文Φ及び自然数ｋを与えて次の(1)～(4)の手続きを行った場合，(4)で手続きを終了すれば，M1，M2はΦに対する生存性を持つ．

［生存性の検証アルゴリズム］

```
procedure LIVENESS(M1,M2,Φ,k) ;
  S←⟨s_I, t_I⟩ ; /* s_I, t_IはM1,M2の初期状態，⟨s_I, t_I⟩はﾏｰｸされていない */
  while S中でﾏｰｸされていない状態対が存在 do
  begin
    ⟨s,t⟩をS中でﾏｰｸされていない状態対とし，Φ及び⟨s,t,ℝn,∇m⟩に対する
     k-到達可能木Tk(⟨s,t,ℝn,∇m⟩,Φ)を作る
    if ⟨s,t⟩がΦに対するk-生存性を持たない
       then print(失敗) ;
    ⟨s,t⟩をﾏｰｸする
    S←S∪{⟨s',t'⟩|⟨s',t'⟩∈Fk(⟨s,t⟩,Φ)
                        且つ⟨s',t'⟩はS中で未だﾏｰｸされていない}
  end ;
  Ω←S ;
  print(成功) ;
end.
```

（証明）略 ■

上述の検証手続きは，高々ｋ回の遷移でΦを満足する状態に遷移可能であるための一



{ *c1 = not(v1 = r2) , *c2 = (v1 = r2) , *c3 = not(r1 >= 2) , *c4 = (r1 >= 2) ,
*c5 = (v1 = r2) , *c6 = not(r3 >= 2) and not(v1 = r2) , *c7 = (r3 >= 2) and not(v1 = r2) }

図３　　2-到達可能木

つの十分条件が成り立つかどうかを調べている．通常，プロトコルマシンを設計する場合，異常な状況に遷移すれば，できるだけ早く正常な状況へ復帰するように仕様が書かれていると考えられるので，kをある程度大きな値にすれば，上述の検証手続きでも実用上生存性の検証が可能であると考えられる．

尚，安全性の証明は，1-生存性の証明と等価である．

6．あとがき

現在，上述の手続きに基づいて安全性や生存性の検証を行うためのプログラムが作成されている．これらのプログラムを用いて，HLDC手順の１次局と２次局の２つのプロトコルマシンを接続した時に安全性や生存性が成り立つことを検証した．これらの結果については文献[3] 参照．また，我々は本論文で用いたモデルを用いて，２つのプロトコルマシンの等価性（互換性）を証明するための方法についても提案している．この方法は本質的に安全性の証明と同様の方法で検証が行われる．詳細については文献[2] 参照．

文　　献

(1) 野口，斉藤，白鳥：”分散処理とコミュニケーション”，ソフトウェア工学ハンドブック（榎本編）オーム社，pp177-221(1986).

(2) 二宮，東野，谷口，木本：”プロトコルマシンの等価性証明の一方法”，信学論(Ｄ)，J71-D，12，pp. 2630-2639(昭63-12).

(3) 木本，東野，谷口，森，二宮：”通信プロトコルにおけるエラーリカバリ性の検証の一方式”，信学技報IN88-80(昭63-09).

(4) J.E.Hopcroft and J.D.Ullman:"Introduction to Automata Theory, Language, and Computation", Addison-Wesley,(1979).

(5) 竹内外史：”数学基礎論の世界”，日本評論社（1972).